Check Design Metric Indicator to Improve Quality Software Development.pdf

by Arta Sundjaja

Submission date: 19-Sep-2019 06:24PM (UTC+0700)
Submission ID: 1102514169
File name: ign_Metric_Indicator_to_Improve_Quality_Software_Development.pdf (857.88K)
Word count: 3510
Character count: 19272

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 135 (2018) 616–623

Procedia Computer Science

www.elsevier.com/locate/procedia

3rd International Conference on Computer Science and Computational Intelligence 2018

Design Metric Indicator to Improve Quality Software Development (Study Case: Student Desk Portal)

Reina Setiawan[a,b,*], Zulfany Erlisa Rasjid[a,b], Andy Effendi[c]

[a]Computer Science Department, School of Computer Science, Bina Nusantara University, K.H. Syahdan No. 9, Jakarta 11480, Indonesia
[b]Computer Science Department, BINUS Graduate Program - Doctor of Computer Science, Bina Nusantara University, Jl. K.H. Syahdan No. 9, Jakarta 11480, Indonesia
[c]Information Systems Department, School of Information Systems, Bina Nusantara University, Jl. K.H. Syahdan No. 9, Jakarta 11480, Indonesia

Abstract

The development of Information Technology and Communications facilitate people running their daily lives. One of the important role in development is application development which is the main element in the development of technology. Due to this reason, it is required to have an application that would satisfy the criteria of standard qualities without any tolerance towards error and failure during process. This paper focused on the design of indicator that can be used in measuring qualities in application development, with Student Desk Portal at Bina Nusantara University used as case study is accessed by thousands of students each week. For this reason, the quality of the application is extremely important. The designed indicators are applied in the development of 35 modules in the Student Desk Portal application. The result shows that after applying the designed indicators in the application development, the number of bugs and errors found in the implementation drastically reduced from 43% to 23%.

Keywords: Quality Software Application; Indicator; Software Metric; Student Desk Portal

1. Introduction

Nowadays, human life cannot be separated from the services of communication and information technology.

* Corresponding author. Tel.: +62-818-818-944.
E-mail address: reina@binus.edu

10.1016/j.procs.2018.08.225

The rapid development of communication and technologies facilitates people in running their lives. One of the important role in development is application development which is the main element in the development of technology. Due to this reason, it is required to have an application that would fulfil the criteria of standard qualities without any tolerance towards error and failure during process. Standard quality criteria of the development of an application involves several aspects, which is reliability, correctness, completeness, efficiency and usability[1]. Beside the criteria described previously, there exists other additional criteria such as adaptability, analysability, clarity, complexity, conciseness, and many other criteria[2].

Previous researches in the field of measuring quality of application includes the impact of code reviews on quality improvement and empirical studies[3,4,5], a model to predict failures from application[6], important role of software application development[7], cyclomatic complexity to calculate the complexity of application[5,6], quality criteria of development that depends on types of applications to be develpoved[2]. A quality metric on application development based on object-oriented designs has also been researched[8].

The purpose of this research is to develop an indicator that can be used in measuring the quality of application development, using Student Desk Portal, Bina Nusantara University. This portal is accessed by thousands of students each week. There are about 10 to 23 modules where errors and bugs are found during implementation. Having a total of 43% errors and/or bugs found, therefore this matter is a problem and becomes an important matter. Attention and care is required during application development and must meet standard quality requirements. Due to this reason, the quality of an application is extremely important. Based on the data that is obtained from the implementation of this student desk portal, For this reason, this research is to handle the existing problems, such that the development of the student desk portal achieves a good quality. The results of this research are metric indicators that should be applied during application development and can also be applied by other application development. The indicators are designed using Goal-Question-Metric (GQM) approach. GQM is an approach to design measurements that provides a clear and solid result because the measurements are rooted from the goal and objectives[9].

2. Literature Review

2.1 Quality Metric of Application Development

The quality of an application is extremely important and is necessary to pay high attention in the process of development. The quality of application development can be measured through quantitative measurement. The criteria includes[10]:

1. *correctness*, *reliability* and *robustness*
2. *performance*
3. *usability*
4. *verifiability*
5. *maintainability* which consists of *reparability and evolvability*
6. *reusability*)
7. *portability*
8. *understandability*
9. *interoperability*
10. *productivity*
11. *timeliness*
12. *visibility*

The choice of the criteria to be measured can be adjusted based on the application which is developed[2].

2.2 Application Development Criteria

Student Desk Portal, which is the object of his research, is an application which is an in-house developed application, supported by more than one database, such as Oracle and MS SQL Server. The usage of this application is thousands of students every week. The selected criteria of the quality indicators design include correctness,

reliability, robustness, verifiability, maintainability that comprises of reparability and evolvability, and interoperability.

The correctness and reliability criteria must be taken into high consideration because this portal is the main portal used by students, in order to support teaching-learning process and other services available at the university, whereas the robustness criteria is required due to its high accessibility. In order to support all of the above criteria, a good testing process is necessary to analyze both formally and informally, up to the testing approaches used. Fig. 2.1 shows the activities of analyzing and testing[11].

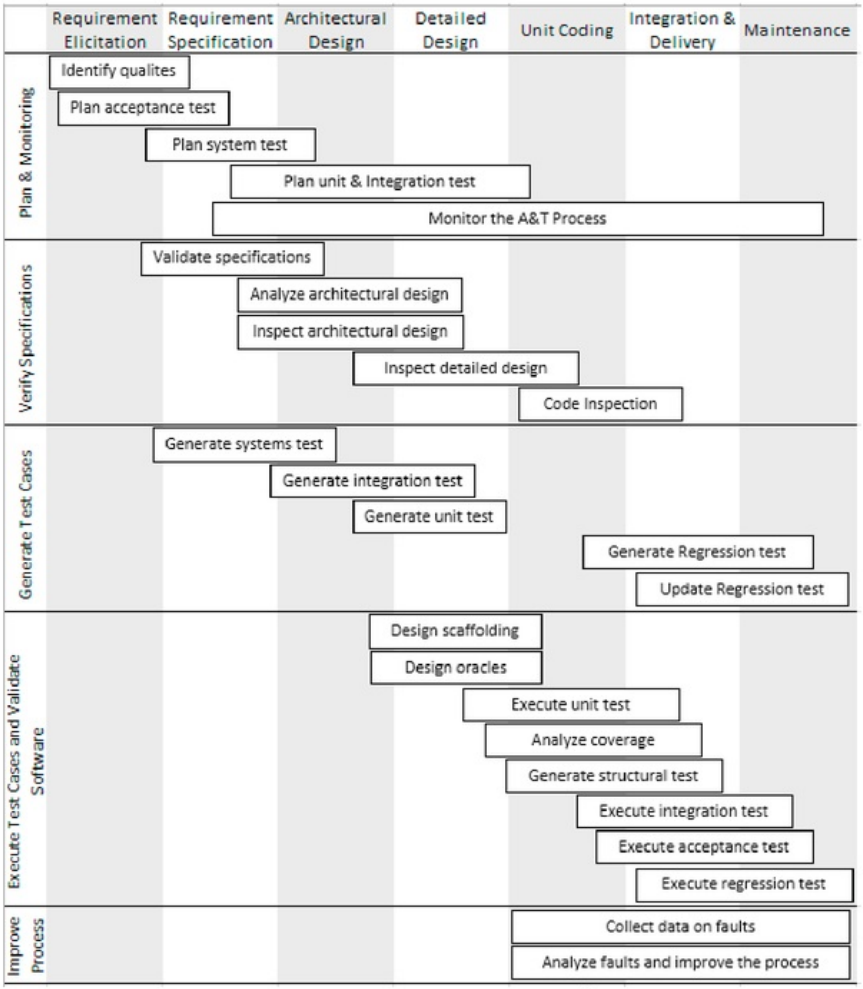

Fig. 2.1. Analysis and testing activities[11]

The maintainability criteria is one of the indicator will be developed based on the number of errors that occurs after implementation. An error of 43% is quite a large number and is critical in relation with the quality of an application which is the intended result of this research.

Another criterion that will be an indicator in this research is interoperability. This is required because the application is supported by more than one data base, such as Oracle and MS SQL Server. The interoperability level of a system is important at implementation of the system, where it is expected to be free or errors and the application can run properly.

2.3 Related Works

Some of the earlier researches related to application development quality metrics is summarized in the following Table 2.1.

Table 2.1. Researches related to Application Development Quality Metrics.

Research	Scope of Research	Method used	Research result
4	The effect of code review in the improvement of quality	Code review using Gerrit tools	Bad Code review gives a negative effect on application development
7	The important role of application development metrics.	Distributing questionnaires to professionals related to the understanding and perception on application process and application measurement processes.	The advantage obtained by applying metrics is the effectiveness of selecting alternatives, efficiencies in the development in terms of use of time and cost, and providing objective information.
2	The relation of software metrics and the application being developed	Comparing all criteria application quality metrics.	Not all application requires the same criteria. Criteria do not only consist of one item but also consist of combination of other criteria.
5	Using software metric in the implementation of an application and improving performance of an application.	Identification of software metric and using cyclomatic complexity to understand the complexity of an application and perform experiments.	Using Java programming language, the quality of an application proven to have an improvement.
6	Reliability analysis conducted in the early stages of development.	Using the indicator software defect density based on fuzzy logic	A model with 4 steps of application development using fuzzy logic
12	Software Quality Factors and Software Quality Metrics to Enhance Software Quality Assurance in Software Development Life Cycle	Combining Software Quality Criteria with Software Metric	Software Quality Metric

Based on the previous research in Table 2.1, there are no general criteria to measure the quality of software development, therefore this research attempt to determine the general criteria for web based application development with Student Desk Portal as the study case.

2.4 Goal-Question-Metric (GQM) Approach

The Goal-Question-Metric (GQM) approach was first introduced by Basili and Weiss in the 1980s and further developed by Rombach. This top-down approach starts by identifying the goal in the first step and then define questions to be answered in order to reach the goal in step two, and then decide on the metrics and preparing a metric schema at step three. Applying GQM allows the development of metrics to become more efficient because the data collected only refers to the data required[13]. The schema of GQM approach consists of 3 steps, which is[9] :

1. Determining the main goal of project development
2. Determining the question of each goal in order to determine whether the questions have been answered.
3. Determining what to be measured in order to answer the questions.

Fig 2.2 shows an example of application using GQM[9] approach.

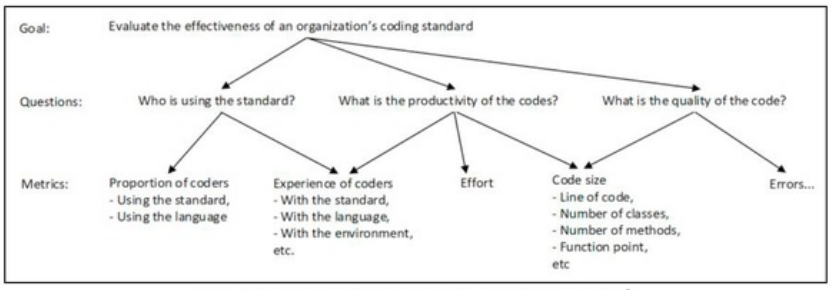

Fig. 2.2 Example of application of GQM approach [9]

The GQM approach has been implemented in several area, such as security metric[14], guidelines in defining non-functional requirement (NFR)[15], evaluating characteristics of software product[16], combination of application development and operational information technology (DevOps)[17] and many others.

3. Research Methods

The research method applied in this research is depicted in the flow diagram in Fig. 3.1

Fig. 3.1. Research Methods Steps

The first step in the research method is the literature study in order to get information and previous works related to the topic in the scope of Software Quality Metric. The research object is Student Portal of Bina Nusantara University. At data collection step, data is collected, which is the data related to the number of bugs from the model existed in the portal. This data is required to provide a comparison with the data resulted from this research, in order to measure the effectiveness.

The next step is to design the indicators to measure the quality in the application development. The indicator design uses the Goal Question Metric (GQM) approach. The indicator selected will be applied to the development of new modules in the student desk portal. The implementation result with the corresponding number of bugs is compared to the number of bugs before implementing the metric.

4. Result and Discussions

There are several criteria to measure the quality of software development[10]. The criteria chosen in this research is related to web-based application development. Based on the indicator criteria selected in order to develop the application, there are seven goals with six questions and seventeen metrics. All of the seven goals with the corresponding six questions and seventeen metrics can be seen in Table 4.1, whereas the mapping from GQM can be seen in Fig.4.1. Table 4.2 explains the quantified evaluation rubric for each and every indicator metrics, evaluated from 1 to 5, where 1 is the lowest possible result and 5 is the best result.

Table 4.1 Goal-Question-Metric (GQM)

Goal #	Goal	Question #	Question	Metric #	Metric
Goal 1	Correctness	Q1	How to ensure that system display the correct information?	M1	Average access to correct database
				M2	Average logic inspection
		Q2	How to ensure that user understands the displayed information?	M3	Number of UAT question (scoring)
Goal 2	Reliability	Q1	How to ensure that system display the correct information?	M4	Average logic inspection
		Q3	How to ensure that the information displayed is valid?	M5	Average data base synchronized inspection
		Q4	How to ensure that user always get the required information?	M6	Average of system failure (infrastructure)
				M7	Average of system failure (server)
				M8	Average of system failure (application)
Goal 3	Robustness	Q4	How to ensure that user always get the required information?	M9	Average of system failure (stress test)
				M10	Average of system failure

Goal #	Goal	Question #	Question	Metric #	Metric
					(application)
Goal 4	Verifiability	Q3	How to ensure that the information displayed is valid?	M11	Number of error during development testing
				M12	Number of errors during UAT
Goal 5	Maintainability	Q4	How to ensure that user always get the required information?	M14	Number of system failures (application) after maintainability
		Q5	How to ensure that maintenance or development of system can be facilitated?	M13	Number of documentation (minimum 1 doc. per module)
Goal 6	Evolvability	Q4	How to ensure that user always get the required information?	M15	Number of system failure (application) after database upgraded
Goal 7	Interoperability	Q6	How to ensure that user always get the required information even though different platform is used?	M16	Number of system failure (application) with different browser
				M17	Number of system failure (application) with different operating system

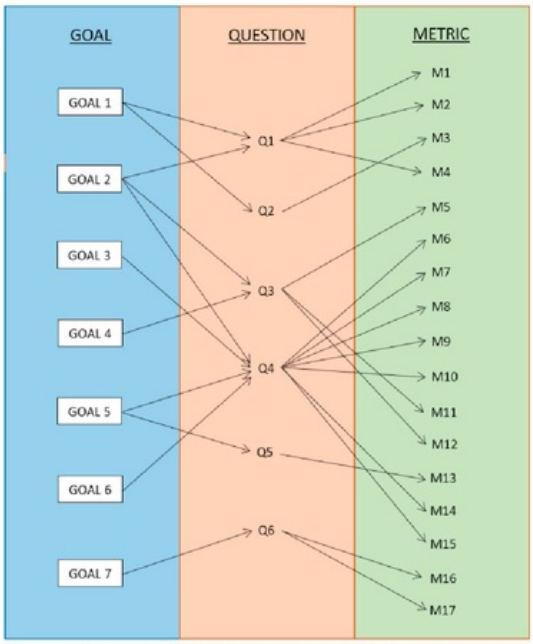

Fig. 4.1 Goal-Question-Metric (GQM) Mapping

The research objects are thirty five modules from the Student Desk Portal, related to exam application which is developed using the indicator criteria that has previously been determined. The thirty five modules are implemented in one and a half months.

Fig 4.2 explains the evaluation results for each module based on the indicators used and the rubric that was determined earlier. From Fig 4.2 it shows that all of the indicators have scores above 4.00 under the scale 1.00 to 5.00. This shows that the development of modules under the control of the selected indicators provide a positive effect on the quality of the resulted module. The result shows nine errors that have to be corrected from eight modules. This shows that the error level is 23% which has decreased significantly compared to 43% in the previous development without considering the indicators. The data explains that applying the indicators that was designed using the GQM approach has shown an improvement in the quality of the application development.

Tabel 4.2 Evaluation Rubric.

Metric #	Metric	Rubric of Score				
		1	2	3	4	5
M1	Average access to correct database	< 60%	60% <= X < 70%	70% <= X < 80%	80% <= X < 100%	100%
M2	Average logic inspection	< 60%	60% <= X < 70%	70% <= X < 80%	80% <= X < 100%	100%
M3	Number of UAT question (scoring)	> 5 comments	3 - 5 comments	2 comments	1 comment	no comment
M4	Average logic inspection	< 60%	60% <= X < 70%	70% <= X < 80%	80% <= X < 100%	100%
M5	Average database synchronized inspection	< 60%	60% <= X < 70%	70% <= X < 80%	80% <= X < 100%	100%
M6	Average of system failure (infrastructure)	> 5 failures	3 - 5 failures	2 failures	1 failure	no failure
M7	Average of system failure (server)	> 5 failures	3 - 5 failures	2 failures	1 failure	no failure
M8	Average of system failure (application)	> 5 failures	3 - 5 failures	2 failures	1 failure	no failure
M9	Average of system failure (stress test)	> 5 failures	3 - 5 failures	2 failures	1 failure	no failure
M10	Average of system failure (application)	> 5 failures	3 - 5 failures	2 failures	1 failure	no failure
M11	Number of error during development testing	> 5 errors	3 - 5 errors	2 errors	1 error	no error
M12	Number of error during UAT	> 5 errors	3 - 5 errors	2 errors	1 error	no error
M14	Number of system failure (application) after maintainability	> 5 failures	3 - 5 failures	2 failures	1 failure	no failure
M13	Number of documentation (minimum 1 doc. per module)	< 60%	60% <= X < 70%	70% <= X < 80%	80% <= X < 100%	100%
M15	Number of system failure (application) after database upgrade	> 5 failures	3 - 5 failures	2 failures	1 failure	no failure
M16	Number of system failure (application) with different browser	> 5 failures	3 - 5 failures	2 failures	1 failure	no failure
M17	Number of system failure (application) with different operating system	> 5 failures	3 - 5 failures	2 failures	1 failure	no failure

Fig. 4.2 Scores based on the evaluation rubric indicator.

Previous works on the use of software metrics to improve quality implements some criteria such as functionality, reliability, maintainability and usability. The result provide an improvement of the application that was developed [18] when those mentioned metrics was used. However, it is based on empirical studies where the specific indicators were not specified in detail. In this paper the criteria is more detailed for each of the quality point measured previously as shown in Table 4.1.

5. Conclusion

The results show that GQM approach is effective to improve quality of software development. In this case, after applying the designed indicators into the development of thirty-five modules in the Student Desk Portal application, the number of errors and bugs obtained in implementation are reduced from 43% down to 23%. The criteria of the indicator that was applied consist of correctness, reliability, robustness, verifiability, maintainability, evolvability, and interoperability. It is concluded that applying quality metrics to application development provide better result eventually.

References

1. Florac W a. Software Quality Measurement: A Framework for Counting Problems, Failures and Faults. C ESC-TE-92-22, Pittsburgh, PA; Softw Eng Institute, Carnegie Mellon Univ Sept. 1992;
2. Chauhan V, Scholar MT. Role of Software Metrics to Improve Software Quality. Int J Comput Sci Inf Technol. 2014;5(3):4167–70.
3. McIntosh S, Kamei Y, Adams B, Hassan AE. The impact of code review coverage and code review participation on software quality: a case study of the qt, VTK, and ITK projects. Proc 11th Work Conf Min Softw Repos - MSR 2014 [Internet]. 2014;192–201. Available at: http://dl.acm.org/citation.cfm?doid=2597073.2597076
4. McIntosh S, Kamei Y, Adams B, Hassan AE. An empirical study of the impact of modern code review practices on software quality. Empir Softw Eng. 2016;21(5):2146–89.
5. Bansal D, Saxena A, Singh G. Improvement and Implementation of Software Quality by Using Software Metrics. Int J Sci Eng Technol [Internet]. 2014;742(36):2277–1581. Available at: http://www.ijset.com/ijset/publication/v3s6/IJSET_2014_610.pdf
6. Yadav HB, Yadav DK. Early software reliability analysis using reliability relevant software metrics. Int J Syst Assur Eng Manag [Internet]. 2014;(1992). Available at: http://link.springer.com/10.1007/s13198-014-0325-3
7. Fernando W, Wijayarathne D, Fernando JSD, Mendis MPL, Guruge I. The Importance of Software Metrics: Perspective of a Software Development Projects in Sri Lanka. Proc SAITM Res Symp Eng Adv Sri Lanka. 2014;2014:91–5.
8. Rosenberg L, Hyatt L. Software quality metrics for object-oriented environments. Crosstalk Journal, April [Internet]. 1997;10(4):1–6. Available at: http://people.ucalgary.ca/~far/Lectures/SENG421/PDF/oocross.pdf
9. Fenton N, Bieman J. Software Metrics A Rigorous and Practical Approach. Third. CRC Press; 2015.
10. Carlo Ghezzi, Mehdi azayeri D, Mandrioli. Fundamentals of Software Engineering [Internet]. Vol. 7141. Prentice-Hall; 2003. Available at: http://link.springer.com/10.1007/978-3-642-29320-7
11. Pezzand M, Young M. Software Testing and Analysis: Process, Principles, and Techniques. John Wiley & Sons; 2008.
12. Ming-Chang L. Software Quality Factors and Software Quality Metrics to Enhance Software Quality Assurance. Br J Appl Sci Technol. 2014;4(21):3069.
13. Koziolek H. Goal, Question, Metric. In: Eusgeld I, Freiling FC, Reussner R, editors. Dependability Metric Advanced Lecturer. Springer-Verlag Berlin Heidelberg; 2008. p. 39–42.
14. Yahya F, Walters RJ, Wills GB. Using Goal-Question-Metric (GQM) Approach to Assess Security in Cloud Storage. In: Enterprise Security. Springer International Publishing; 2017. p. 223–40.
15. Silva A, Pinheiro PR, Albuquerque A, Barroso J. Evaluation of an approach to define elicitation guides of non-functional requirements. IET Softw [Internet]. 2017;11(5):221–8. Available at: http://digital-library.theiet.org/content/journals/10.1049/iet-sen.2016.0302
16. Calabrese J, Muñoz R, Pasini A, Esponda S, Boracchia M, Pesado P. Assistant for the Evaluation of Software Product Quality Characteristics Proposed by ISO/IEC 25010 Based on GQM-Defined Metrics. In: De Giusti AE, editor. Computer Science – CACIC 2017. Springer International Publishing; 2018. p. 164–75.
17. König L, Steffens A. Towards a Quality Model for DevOps. Contin Softw Eng Full-scale Softw Eng. 2018;37.
18. Venkataraman S, Hussein A Al. Development of software metrics for improving the quality of the under graduate student projects in computer science / information science / information technology / computer engineering. Int J Comput Sci Netw Secur [Internet]. 2017;17(10):212–22. Available at: http://paper.ijcsns.org/07_book/201710/20171027.pdf

Check Design Metric Indicator to Improve Quality Software Development.pdf

ORIGINALITY REPORT

9%	6%	8%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	stuartgeiger.com Internet Source	3%
2	Cuk Tho, Arden S. Setiawan, Andry Chowanda. "Forming of Dyadic Conversation Dataset for Bahasa Indonesia", Procedia Computer Science, 2018 Publication	2%
3	Marianus Magno Seran Bria, Gede Putra Kusuma, Louis Khrisna Putera Suryapranata. "PROMOTING TIMOR LESTE'S TAIS CLOTH USING MOBILE AUGMENTED REALITY APPLICATION", Procedia Computer Science, 2018 Publication	2%
4	uad.portalgaruda.org Internet Source	1%
5	Submitted to University of the West Indies Student Paper	1%
6	Shyama Prasad Mukherjee. "Chapter 16 Reliability—An Extension of Quality", Springer Nature, 2019	1%

Publication

7	Submitted to Middle East University Student Paper	1%

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		